HW #2

1. Create a C++ program that analyzes a predefined paragraph of text stored within your code as a constant character array. The program should output the number of sentences, the average number of words per sentence (rounded down to the nearest whole number), and the frequency of vowels (a, e, i, o, u) throughout the paragraph. A sentence is defined as any string of characters that ends with a period (.), exclamation mark (!), or question mark (?). A word is any string of letters delimited by whitespace, punctuation, or the beginning/end of a sentence. Assume the input consists only of letters, whitespace, and punctuation (.,!?), and that sentences are well-formed.

   When counting vowels, consider upper- and lowercase versions as the same. The output should include the total number of sentences, the average number of words per sentence, and the count of each vowel that occurs, displayed in alphabetical order. Only list vowels that appear in the paragraph.

   Example:
   Given the predefined paragraph in your code:

   ```
   const char paragraph[] = "Hello, world! I am learning C++. Isn't it exciting?";
   ```

   The expected output should be:

   ```
   3 sentences
   Average 4 words per sentence
   4 a
   6 e
   3 i
   1 o
   ```

   Technical Requirements

   - After implementing the analysis functionality, demonstrate the solution by running your program with both predefined paragraphs. The program should output the analysis results for each paragraph separately. Your program's main function should look something like this in structure:

```
int main() {
        analyzeText(paragraph1);
        std::cout << "----------------------------------\n";
        analyzeText(paragraph2);
        return 0;
}
```

- Define the paragraph as a constant character array in your program.
- Use functions to break down the problem into manageable tasks, such as counting sentences, counting words, and counting vowels. Pass character arrays and counters via pointers when necessary.
- **Using the C++ Standard Library's string class and functions is NOT allowed. Focus on manipulating the character array directly using pointers and array indexing.**
- Write your program to perform analysis directly on this constant character array without reading input from stdin.
- Demonstrate control flow, loop constructs, and pointer arithmetic in your solution.
- Strive to implement your code adhering to industry-standard coding practices and principles. This includes clear documentation, meaningful variable names, modular design, efficient use of data structures, and adherence to the DRY (Don't Repeat Yourself) principle. Your code should be well-organized, easy to read, and maintainable, reflecting a high level of professionalism expected in the software development industry.